



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Digital Channels Private Edition Guide

Enable a tenant for Digital Channels

9/19/2024

Contents

- 1 Prerequisites
- 2 Get contact center ID from GWS
- 3 Add GWS to the list of available GWS services for Nexus
 - 3.1 Create the authentication client
 - 3.2 Verify client
 - 3.3 Add GWS to the nex_gapis table for Nexus
- 4 Provision Digital Channels in GWS
- 5 Get the authentication token
- 6 Provision the Universal Contact Service connection
- 7 Provision the tenant in Digital Channels
- 8 Enable routing using Designer Applications

Learn how to enable your tenant for Digital Channels.

Related documentation:

-
-
-

RSS:

- [For private edition](#)

Important

“Nexus” is the simplified name we use for the Digital Channels application and nodes, so you’ll see that name referenced throughout this document.

Complete the steps on this page to provision your tenant and set up Digital Channels to work with Web Services and Applications (GWS).

Prerequisites

Review the **Prerequisites** table and make sure you have all the listed information before you get started. The values in this table are referenced later by the name in the Variable column.

Prerequisites

| Parameter | Variable | Type | Example | Notes |
|---------------------------|----------------------|------------------|--------------------------------------|--|
| GWS URL | \$gwsURL | HTTPS URL string | https://gws.mydomain.com | Host of the Web Services and Applications load balancer. |
| Auth URL | \$authURL | HTTPS URL string | https://auth.mydomain.com | Host of the Auth Services load balancer. |
| GWS admin username | \$gwsAdminName | string | ops | The GWS ops credentials. See Provision Genesys Web Services and Applications. |
| GWS admin password | \$gwsAdminPass | string | ops | The GWS ops credentials. See Provision Genesys Web Services and Applications. |
| Tenant User | \$tenantUser | string | nexus | The Person username from your tenant configuration. See Create a Person object. |
| Tenant User Password | \$tenantUserPassword | string | nexus_password | The Person password from your tenant configuration. See Create a Person object. |
| Contact Center ID | \$ccId | string | 578ec98e-f07c-46ad-9675-f36c22f3ba9f | The contact center ID provisioned in Web Services and Applications. If you don't have this ID, see Get contact center ID from GWS below. |
| Tenant Name | \$tenantName | string | premise_tenant | The tenant name you used as in the Enable Nexus UX step. |
| Digital Channels FQDN URL | \$nexusURL | HTTPS URL string | https://nexus.mydomain.com | The fully qualified domain |

| Parameter | Variable | Type | Example | Notes |
|-----------|-----------------------|------------------|---------------------------------------|--|
| | | | | name (FQDN) for Digital Channels. |
| UCS URL | <code>\$ucsURL</code> | HTTPS URL string | <code>https://ucs.mydomain.com</code> | A URL pointing on the Universal Contact Service (UCS) entry point. |

Important

Use a REST client or curl utility to make the requests explained on this page. Make sure to substitute the variables - prefixed with '\$' - with their values.

Get contact center ID from GWS

If you don't have the contact center ID (as specified in the **Prerequisites** table above), you can get it with the following request:

```
curl --user $gwsAdminName:$gwsAdminPass --request GET '$authURL/environment/v3/contact-centers'
```

The expected response:

```
{
  "data": {
    "contactCenters": [
      {
        "id": "",
        "environmentId": "",
        "domains": [
        ],
        "auth": "configServer"
      }
    ]
  }
}
```

Add GWS to the list of available GWS services for Nexus

Complete the steps in this section to add GWS to the list of available GWS services for Nexus.

Create the authentication client

To create the authentication client, send a POST request to GWS that includes a body parameter called **data** in JSON format. Give **data** the following properties:

| Property | Value |
|----------------------|---|
| clientType | CONFIDENTIAL |
| internalClient | true |
| authorizedGrantTypes | refresh_token, password, client_credentials, authorization_code |
| redirectURIs | \$nexusURL |

| Property | Value |
|-------------------------------|--|
| authorities | ROLE_INTERNAL_CLIENT |
| scope | * |
| description | nexus_client |
| name | nexus_client |
| client_id | nexus_client |
| client_secret | - randomly generated and saved for further use |
| accessTokenExpirationTimeout | 43200 |
| refreshTokenExpirationTimeout | 2592000 |

Sample request

```
curl --user ops:ops --request POST '$gwsURL' \  
--header 'Content-Type: application/json' \  
--data '{"data": {  
  "internalClient": true,  
  "name": "nexus_client",  
  "clientType": "CONFIDENTIAL",  
  "client_id": "nexus_client",  
  "client_secret": "",  
  "authorities": ["ROLE_INTERNAL_CLIENT"],  
  "scope": ["*"],  
  "authorizedGrantTypes": ["client_credentials", "authorization_code", "refresh_token", "implicit", "password"],  
  "redirectURIs": ["$nexusURL"],  
  "accessTokenExpirationTimeout": 43200,  
  "refreshTokenExpirationTimeout": 2592000  
}'
```

In the above example, the `ops:ops` GWS superuser is different for your environment.

Expected response

The expected response is **200 OK**.

Verify client

To verify that authentication was successful, send a POST request to GWS:

```
curl --user nexus_client: \
--request POST '$authURL/auth/v3/oauth/token?grant_type=client_credentials>ope=*&client_id=nexus_client&client_secret='
```

This is the response:

```
{"access_token":"","token_type":"bearer","expires_in":43199,"scope":"*"}
```

Add GWS to the nex_gapis table for Nexus

Execute the following query in the PostgreSQL command line interface:

```
INSERT INTO nex_gapis (url, clientid, apikey, clientsecret, created) VALUES ('$gwsURL', 'nexus_client', 'NA', '', now());
```

Provision Digital Channels in GWS

To provision Digital Channels in GWS, send a POST request to GWS:

```
curl --user $gwsAdminName:$gwsAdminPass --request POST '$authURL/environment/v3/contact-centers/$ccId/settings' \
--header 'Content-Type: application/json' \
--data '{
  "data":
  {
    "name": "chatservice-uri",
    "location": "/",
    "value": "$nexusURL",
    "shared": false
  }
}
```

Get the authentication token

To get the authentication token, send a POST request to GWS:

```
curl --user nexus_client: --request POST '$authURL/auth/v3/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'username=$ccId\$tenantUser' \
--data-urlencode 'client_id=nexus_client' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'password=$tenantUserPassword'
```

The expected response:

```
{"access_token": "", "token_type": "bearer", "refresh_token": "", "expires_in": 43199, "scope": "*"}
```

As an output of this step, you will have the GWS access token:

| Parameter | Variable | Type | Example | Notes |
|---------------|---------------------|-------------|--------------------------------------|---|
| Nexus API Key | \$outGWSAccessToken | UUID string | 9b7682b7-cbce-422f-90bb-ecda85e61695 | The access_token from the response example. |

Provision the Universal Contact Service connection

You must complete this step if you are using Universal Contact Service in your environment.

Make the following POST request to create the PlatformUCS service:

```
curl -X POST \
  $nexusURL/nexus/v3/provisioning/services/$ccid/PlatformUCS \
  -H 'Content-Type: application/json' \
  -H 'x-api-key: $apiKey' \
  -H 'x-ccid: $ccid' \
  -d '{
    "url" : "$ucsURL",
    "secret": {},
    "data" : {}
  }'
```

Provision the tenant in Digital Channels

To provision the tenant in Digital Channels, make the following POST request:

```
curl --request POST '$nexusURL/nexus/v3/provisioning/tenants' \
--header 'Authorization: Bearer $outGWSAccessToken' \
--header 'Content-Type: application/json' \
--data '{
  "id": "$ccId",
  "name": "$tenantName",
  "type": "PureEngage",
  "backendurl": "$gwsURL",
  "username": "$tenantUser",
  "token": "$tenantUserPassword",
  "genesystemenantid": 1
}'
```

The expected response:

```
{
  "status": {
    "code": 0
  },
  "data": {
    "xapikey": "$nexusApiKey"
  },
  "operationId": "ec90f3d2-f4b5-47fd-9004-2c309398ab38"
}
```

As an output of this step, you will have the specific tenant API key:

| Parameter | Variable | Type | Example | Notes |
|---------------|---------------|-------------|--------------------------------------|--|
| Nexus API Key | \$nexusApiKey | UUID string | 9b7682b7-cbce-422f-9bb0-ecda85e61695 | The xapikey from the response example. For creating chat. |

Enable routing using Designer Applications

Use Agent Setup to add the following values to the "DesignerEnv" transaction in the Environment tenant. If the transaction doesn't exist, create it with a type of List and "DesignerEnv" as Name and Alias.

| Property | Value |
|----------|-----------|
| baseurl | /nexus/v1 |
| password | |
| url | /nexus/v3 |